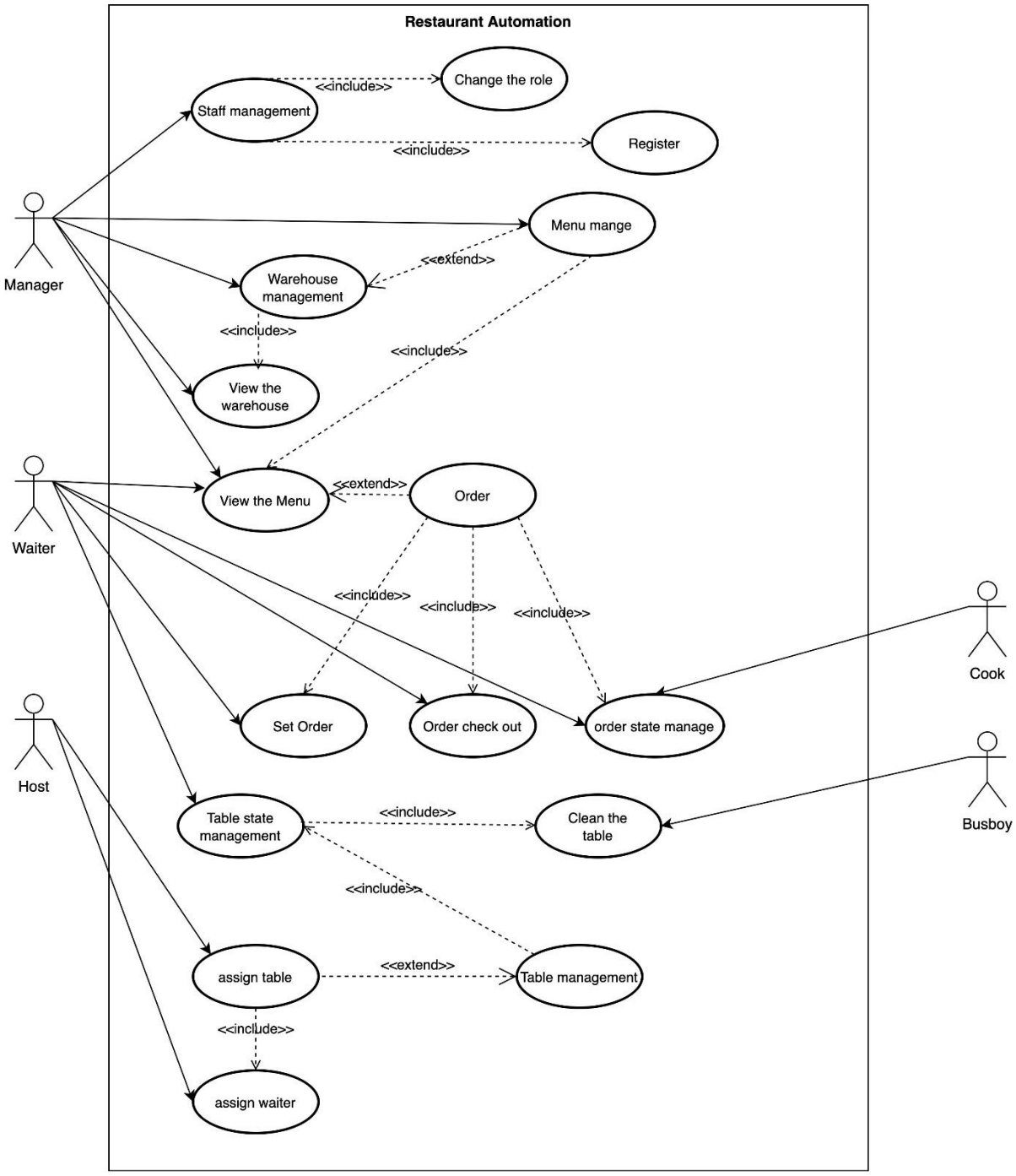
Software Quality

Deliverable 1. 2

Team Spark 315

Requirement

A. Use case diagram



B. Use case elaboration

Use Case	Staff management
Actors	Manager
Goal	Manage the information and state of employees
Preconditions	Manager has the permission and total information of employees
Scenarios	Employee's old and new information is provided Change employee's information
Exceptions	Employee's new information

Use Case	Change the role
Actors	Manager
Goal	Temporary change the level of employees
Preconditions	Employee does exist
Scenarios	Confirm the permission of manager Confirm the target employee's level Change the employee's level temporary Wait until the time limit arrive Change back the employee's level
Exceptions	Time limitation is given Target level is in permit Target employee's information is existing

Use Case	Register
Actors	Manager
Goal	Add new employees to database
Preconditions	Employee does not exist
Scenarios	Employee's information is provided Employee is created Employee's level has been confirmed Employee's salary has been confirmed
Exceptions	Unique id, password, job information and level

Use Case	Menu manage
Actors	Manager
Goal	Manage the information and state of menu
Preconditions	Operator has the permission
Scenarios	Confirm the menu state Check the new dishes give or the dishes need to be deleted Add or remove the dishes Save the new menu and upload it
Exceptions	New dishes do not exist in the menu The dishes want to be removed from menu The given dishes have typical and correct information

Use Case	Warehouse management
Actors	Manager
Goal	Add items to the warehouse Check the remaining of the items in the warehouse Delete items from the warehouse
Preconditions	The item dose not available on the warehouse
Scenarios	The user input all the item that they have in the restaurant The menu will change according the warehouse remaining
Exceptions	Quantity and items do not match Quantity and items need to be provided at the same time

Use Case	View the warehouse
Actors	Manager
Goal	Check the remaining of the items in the warehouse
Preconditions	The item should be available on the warehouse
Scenarios	According the remaining to change the menu
Exceptions	Invalid operation

Use Case	View the menu
Actors	Waiter
Goal	Check the item of the menu Add dishes to the order
Preconditions	The item should be existing on the menu
Scenarios	Check the item the customer wants to order Add dishes to order
Exceptions	The item should be unique

Use Case	Order
Actors	Waiter, Cook
Goal	Check out the order Set the order Check the order detail Check the order state Change the order state
Preconditions	Order state can be operated by waiter and cook Order should exist
Scenarios	Waiter set the order to the kitchen When the dished finish, the order is finish
Exceptions	Invalid operation

Use Case	Set Order
Actors	Waiter
Goal	Order dishes for tables
Preconditions	Customers have decided to order when waiter at table

Scenarios	Waiters help customers order dishes via their devices
Exceptions	Set an empty order

Use Case	Order check out
Actors	Waiter
Goal	Finish orders for tables
Preconditions	Customers finish eating and waiting for check out
Scenarios	Waiters help customers check out
Exceptions	Order dose not check out then assign next round of customer

Use Case	Order state manage
Actors	Waiter and Cook
Goal	Change status at each step of orders
Preconditions	Both waiter and cook can have permission to manage the status of orders, they need to act when the status of orders has changed
Scenarios	Waiters can change the status of each orders when the customers have ordered. Cook can change orders when the dishes are ready, and the orders are going to be paid after checking out by waiters.
Exceptions	Can not be cancel while the dishes

Use Case	Table state management
Actors	Waiter & Busboy
Goal	Managers and staffs (waiters, hosts and busboy) can get the current status of tables
Preconditions	Waiters and busboy have the admission to change tables' status
Scenarios	Change the empty table to be used
Exceptions	Invalid operation

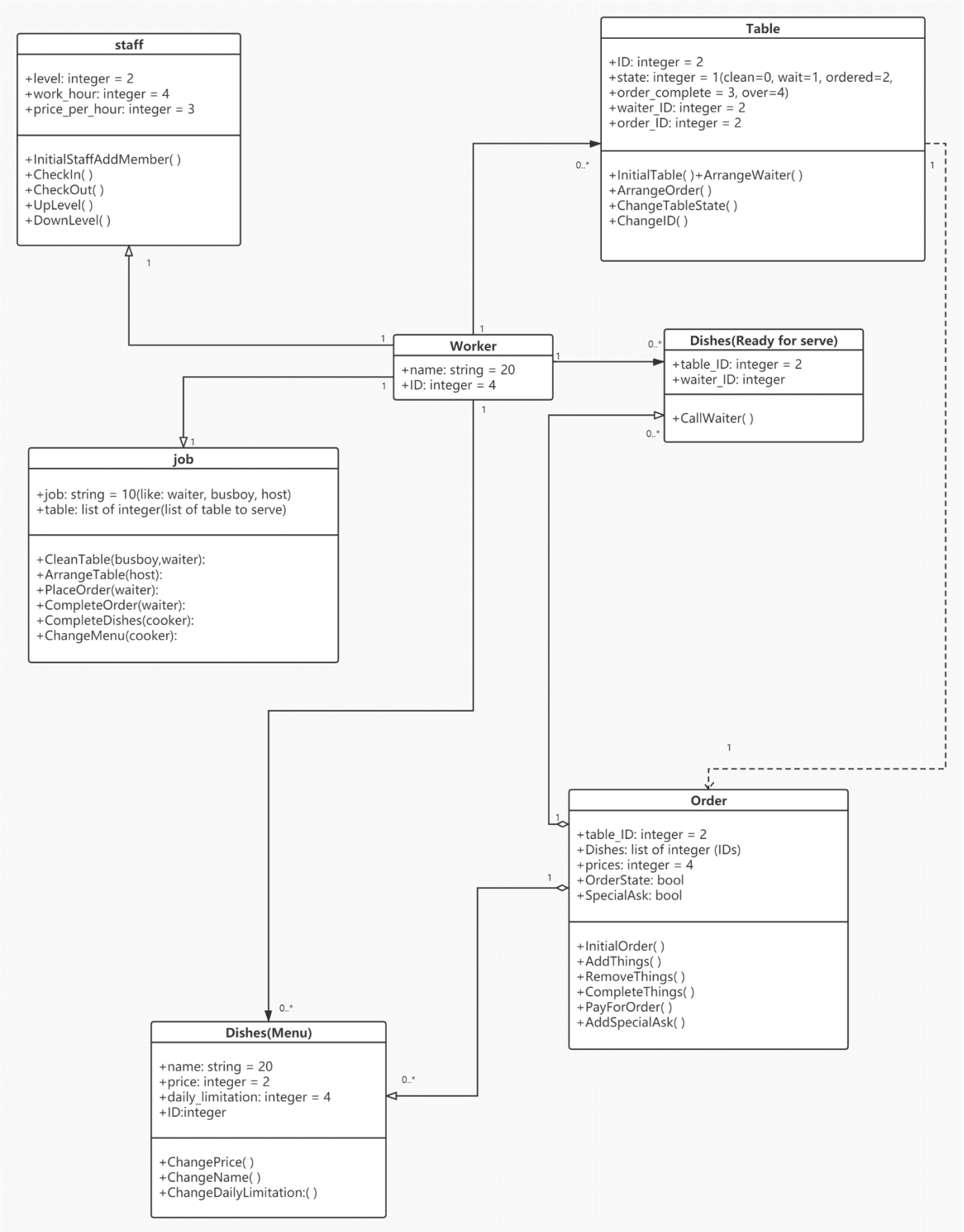
Use Case	Assign table
Actors	Host
Goal	Assigning available table to customers
Preconditions	The tables are empty, and the busboys have cleaned the table
Scenarios	The hosts check their devices and make sure the tables are available.
Exceptions	Assign unclean table for the customers

Use Case	Assign waiter
Actors	Host
Goal	Assigning a waiter to customers
Preconditions	Similar number of tables served by each waiter
Scenarios	The hosts check their devices and make sure the waiters have similar number of tables to serve.
Exceptions	One waiter serves too many customers

Use Case	Table management
Actors	Waiter, Host, Busboy
Goal	Manage the tables state (Occupy, Vacant, Dirty)
Preconditions	Three states as a cycle. It means starting with the vacant to occupy and ending with the dirty.
Scenarios	When the tables are in vacant state, the hosts can assign these table to customers, then change the state to occupy. After the customers check out, the waiters can change occupy state to dirty.
Exceptions	Setting the tables in the wrong state

Use Case	Clean the table
Actors	Busboy
Goal	Cleaning the dirty tables and change their state from dirty to vacant.
Preconditions	After the customers used the tables and checked out.
Scenarios	Busboy clean the tables which show dirty in their devices.
Exceptions	Setting the tables in the wrong state

C. UML Class Diagram



Quality Plan

A. Quality goals and metrics

Product Quality	Quality Goals	Quality Metrics	Strategy
Availability	This system continuously works during business hours	During business hours, this system has 99% availability	Azure deployments automated testing
Reliability	This system can work correctly at all business hours	This system can accept less than 5% error occur	Workload stress testing
Robustness	This system can process all inputs from different types of inputs	There is no error, depending on the input data and data acquisition sequence	Testing different kind of input data and uicode
Learnability	This system can operate intuitively.	A new user can understand this system by seeing its display	Remote usability testing
Usability	This system is extreme simple for users to use.	This system interface has clear and direct operation display	Remote usability testing
Efficiency	This system has a clear and quick response	The system should return search results in < 1s at the 95th percentile	Unit test on normal cases and boundary cases
Security	The system should not be compromised	Monthly audits should reveal fewer than 3 security issues per KLOC	Unit test on normal cases and boundary cases
Portability	This system can work for different restaurant configuration. The setting time most less than half hours.	The system should work with current major database technologies	SQL test
Process Quality			
Maintainability	This system can maintain easily	The source code should not contain lexical and design anti-patterns, be well documented, readable, and not contain any code smells	Code Review, Static code analysis tools
Testability	This system can be tested easily	Every class must have both white and black box testing, with integration /mutation testing where possible	Unit and mutation testing tools

Priority of quality goals: (high->low)

1. Availability, reliability
2. Learnability, usability
3. Efficiency
4. Robustness, portability
5. Maintainability, testability
6. Security

Additional notes:

It is impossible to complete each project in a semester, and to make everything perfect. After discussion, we ranked the importance of quality objectives. Because this is a restaurant automation, our primary goals are **availability** and **reliability**. The most important thing for a software is that when the user opens it, the function must be available and the software can be opened. This is the most basic and most important thing, otherwise other goals cannot be achieved. The second most important thing is **learnability and usability**. The restaurant service staff for this system is very important for them to get started quickly and easily. So, the software needs to provide a friendly experience. This allows employees to spend less time learning and using the software. **Efficiency** is the same reason as above.

Robustness and portability are the goals to be achieved next, as restaurant staff may use the system in a fast and tense work environment, and there may be operating errors. At this time, the system must make effective measures to prevent this kind of thing from happening. The rest are the **maintainability, testability and security**. These qualities are not directly observed by end users, and we expect limited future development of the application. They are important because they better enable developers to better achieve other quality goals.

B. Costs of quality

Task Name	Estimated Effort (hrs)	Implementation	Evaluation	Prevention	Rework
Project Planning	5	3	1	0	1
Infrastructure setup	5	3	1	0	1
UI Design	5	2	2	0	1
Initial Project development and Unit Testing	40	20	5	5	10
Quality Assurance-Integration/E2E testing	20	10	3	0	7
Acceptance Testing	10	5	3	1	2
Software Bug Correction	15	5	4	0	1
Total	**100**	**48**	**19**	**6**	**23**